



US009253239B2

(12) **United States Patent**  
**Levanon et al.**

(10) **Patent No.:** **US 9,253,239 B2**  
(45) **Date of Patent:** **\*Feb. 2, 2016**

(54) **OPTIMIZED IMAGE DELIVERY OVER LIMITED BANDWIDTH COMMUNICATION CHANNELS**

(71) Applicant: **BRADIUM TECHNOLOGIES LLC**, Suffern, NY (US)

(72) Inventors: **Isaac Levanon**, Raanana (IL); **Yonatan Lavi**, Raanana (IL)

(73) Assignee: **BRADIUM TECHNOLOGIES LLC**, Suffern, NY (US)

(\*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 0 days.

This patent is subject to a terminal disclaimer.

(21) Appl. No.: **14/547,148**

(22) Filed: **Nov. 19, 2014**

(65) **Prior Publication Data**

US 2015/0180928 A1 Jun. 25, 2015

**Related U.S. Application Data**

(63) Continuation of application No. 13/027,929, filed on Feb. 15, 2011, now Pat. No. 8,924,506, which is a continuation-in-part of application No. 12/619,643, filed on Nov. 16, 2009, now Pat. No. 7,908,343, which

(Continued)

(51) **Int. Cl.**  
**G06F 15/16** (2006.01)  
**H04L 29/06** (2006.01)

(Continued)

(52) **U.S. Cl.**  
CPC ..... **H04L 65/602** (2013.01); **G06F 3/1454** (2013.01); **G06T 3/4092** (2013.01); **H04L 67/42** (2013.01); **G09G 2340/02** (2013.01); **G09G 2350/00** (2013.01)

(58) **Field of Classification Search**

CPC ..... G06T 3/4092; G09G 2340/02; G06F 3/14  
USPC ..... 709/202, 203, 217, 230, 246, 247;  
345/625; 382/232, 305

See application file for complete search history.

(56) **References Cited**

**U.S. PATENT DOCUMENTS**

4,682,869 A \* 7/1987 Itoh et al. .... 358/426.12  
4,972,319 A 11/1990 Delorme

(Continued)

**FOREIGN PATENT DOCUMENTS**

EP 1070290 B1 1/2001  
WO 99/41675 8/1999

**OTHER PUBLICATIONS**

Potmesil Maps Alive with dArnaud Declaration (all pages).

(Continued)

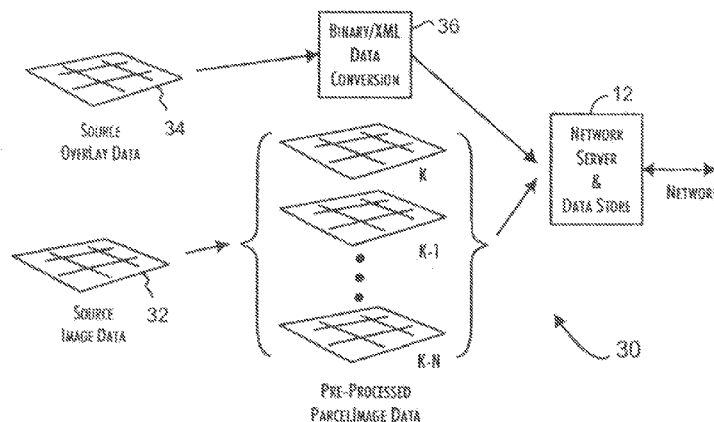
*Primary Examiner* — David Lazaro

(74) *Attorney, Agent, or Firm* — Anatoly S. Weiser, Esq.; Techlaw LLP

(57) **ABSTRACT**

Large-scale images are retrieved over network communications channels for display on a client device by selecting an update image parcel relative to an operator controlled image viewpoint to display via the client device. A request is prepared for the update image parcel and associated with a request queue for subsequent issuance over a communications channel. The update image parcel is received from the communications channel and displayed as a discrete portion of the predetermined image. The update image parcel optimally has a fixed pixel array size, is received in a single and or plurality of network data packets, and were the fixed pixel array may be constrained to a resolution less than or equal to the resolution of the client device display.

**25 Claims, 5 Drawing Sheets**



**Related U.S. Application Data**

is a continuation of application No. 10/035,987, filed on Dec. 24, 2001, now Pat. No. 7,644,131.

- (60) Provisional application No. 60/258,465, filed on Dec. 27, 2000, provisional application No. 60/258,466, filed on Dec. 27, 2000, provisional application No. 60/258,467, filed on Dec. 27, 2000, provisional application No. 60/258,468, filed on Dec. 27, 2000, provisional application No. 60/258,488, filed on Dec. 27, 2000, provisional application No. 60/258,489, filed on Dec. 27, 2000.

- (51) **Int. Cl.**  
**G06F 3/14** (2006.01)  
**G06T 3/40** (2006.01)

- (56) **References Cited**

**U.S. PATENT DOCUMENTS**

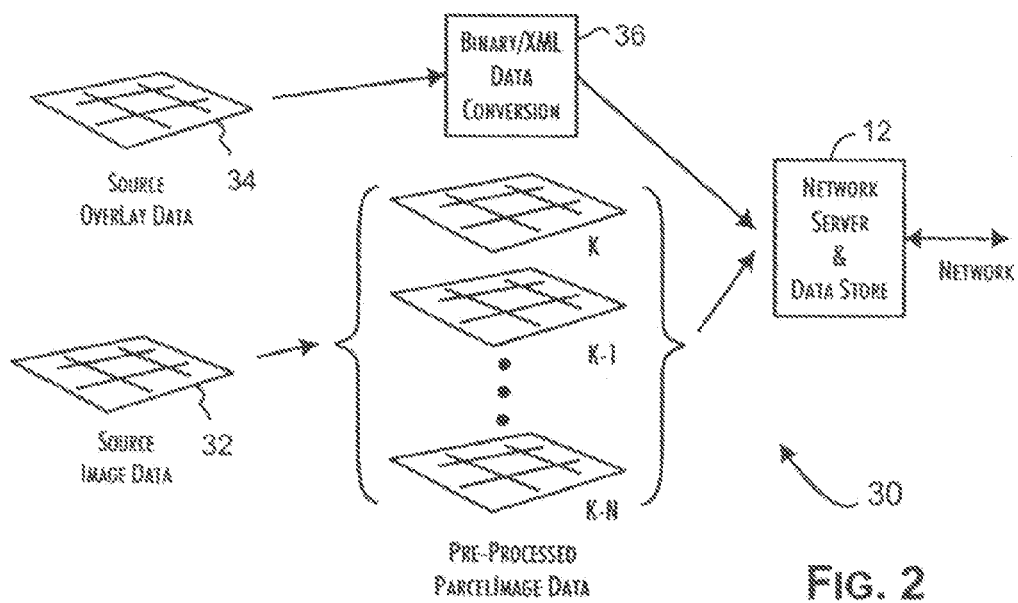
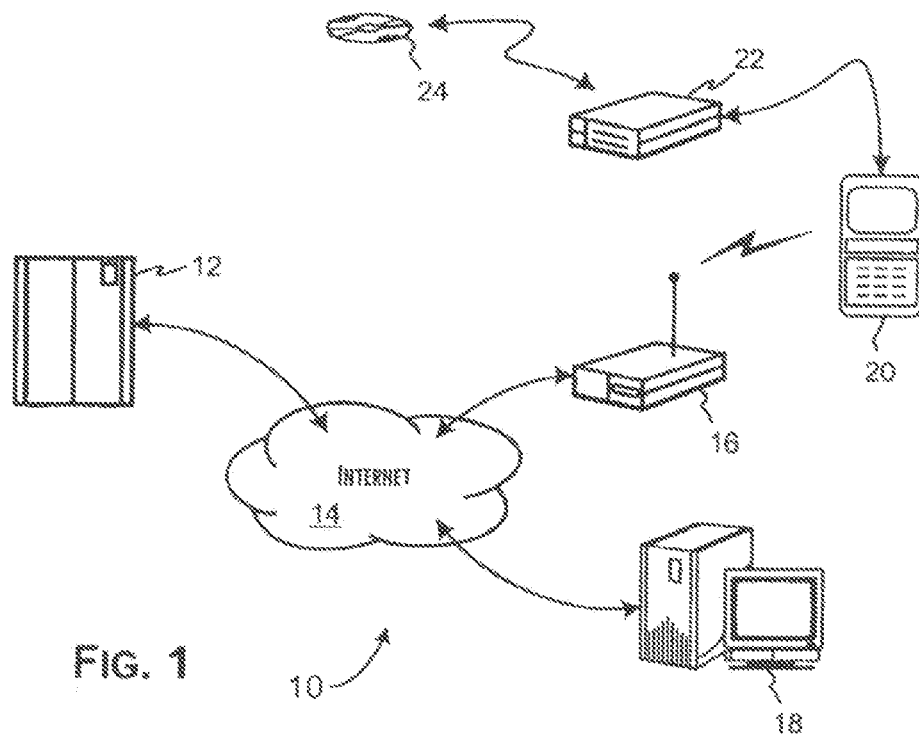
5,179,638	A	1/1993	Dawson et al.	
5,263,136	A	11/1993	DeAguiar et al.	
5,321,520	A	6/1994	Inga et al.	
5,559,936	A *	9/1996	Poulter et al.	345/428
5,613,051	A *	3/1997	Iodice et al.	345/428
5,682,441	A	10/1997	Ligtenberg et al.	
5,760,783	A	6/1998	Migdal	
5,798,770	A	8/1998	Baldwin	
5,940,117	A	8/1999	Hassan et al.	
5,987,256	A	11/1999	Wu et al.	
5,995,903	A *	11/1999	Smith et al.	701/470
6,118,456	A	9/2000	Cooper	
6,167,442	A *	12/2000	Sutherland et al.	709/217
6,182,114	B1	1/2001	Yap et al.	
6,212,301	B1 *	4/2001	Warner et al.	382/232
6,246,797	B1 *	6/2001	Castor et al.	382/232
6,285,317	B1 *	9/2001	Ong	342/357.57
6,314,452	B1 *	11/2001	Dekel et al.	709/203
6,326,965	B1 *	12/2001	Castelli et al.	345/420
6,345,279	B1 *	2/2002	Li et al.	
6,346,938	B1 *	2/2002	Chan et al.	345/419
6,397,259	B1 *	5/2002	Lincke et al.	709/236
6,449,639	B1 *	9/2002	Blumberg	709/217
6,496,189	B1 *	12/2002	Yaron et al.	345/428
6,525,732	B1 *	2/2003	Gadh et al.	345/428
6,608,628	B1 *	8/2003	Ross et al.	345/619
6,608,933	B1 *	8/2003	Dowell et al.	382/232
6,625,309	B1 *	9/2003	Li et al.	382/173
6,650,998	B1	11/2003	Rutledge et al.	
6,704,024	B2 *	3/2004	Robotham et al.	345/581
6,704,791	B1 *	3/2004	Harris	709/231
6,711,297	B1 *	3/2004	Chang et al.	382/240
6,754,365	B1 *	6/2004	Wen et al.	382/100
6,801,665	B1 *	10/2004	Atsumi et al.	382/239
6,882,755	B2 *	4/2005	Silverstein et al.	382/282
6,898,311	B2 *	5/2005	Whitehead	382/166
6,970,604	B1 *	11/2005	Chai	382/240
7,908,343	B2 *	3/2011	Levanon et al.	709/217
8,924,506	B2 *	12/2014	Levanon et al.	709/217

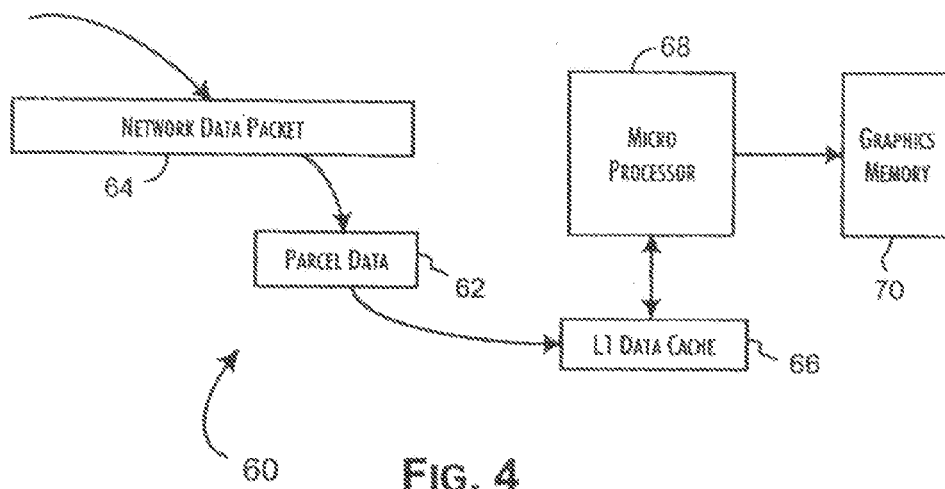
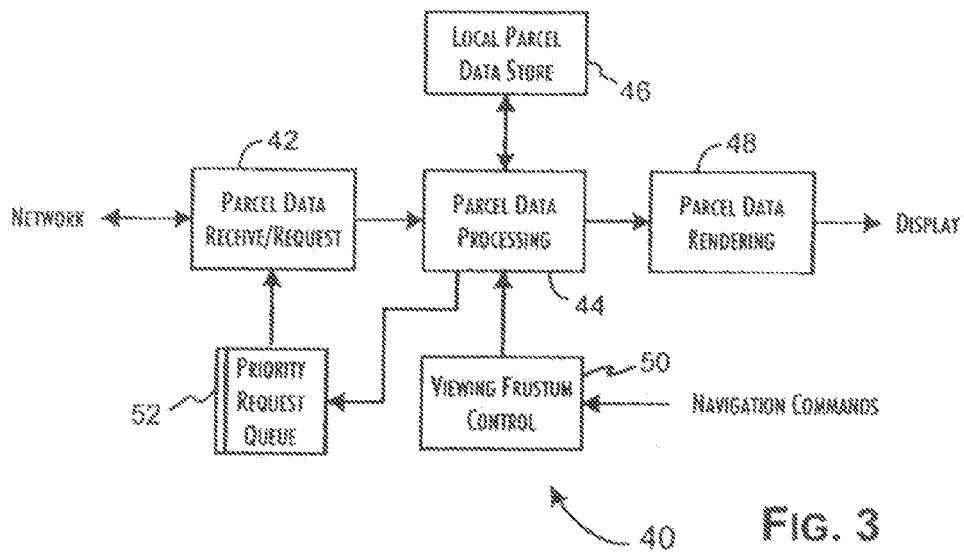
**OTHER PUBLICATIONS**

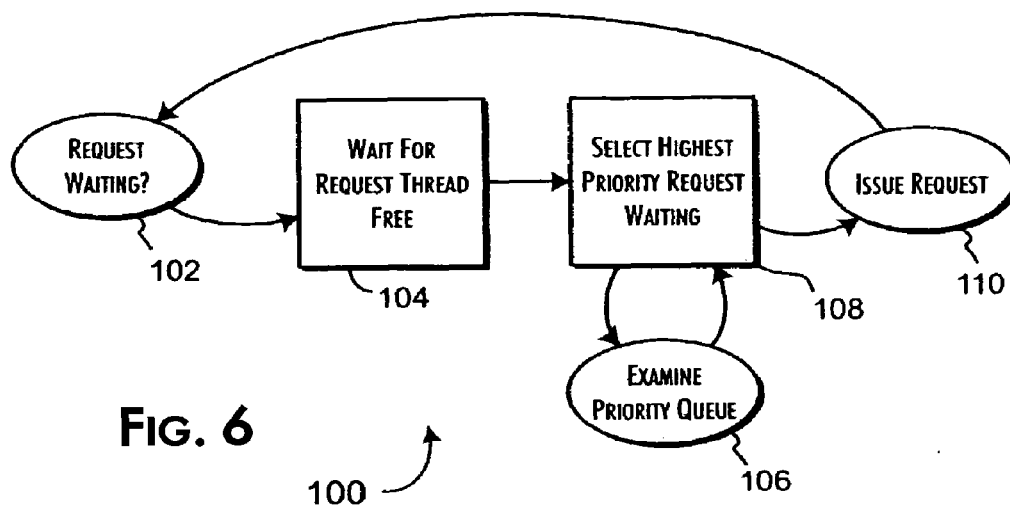
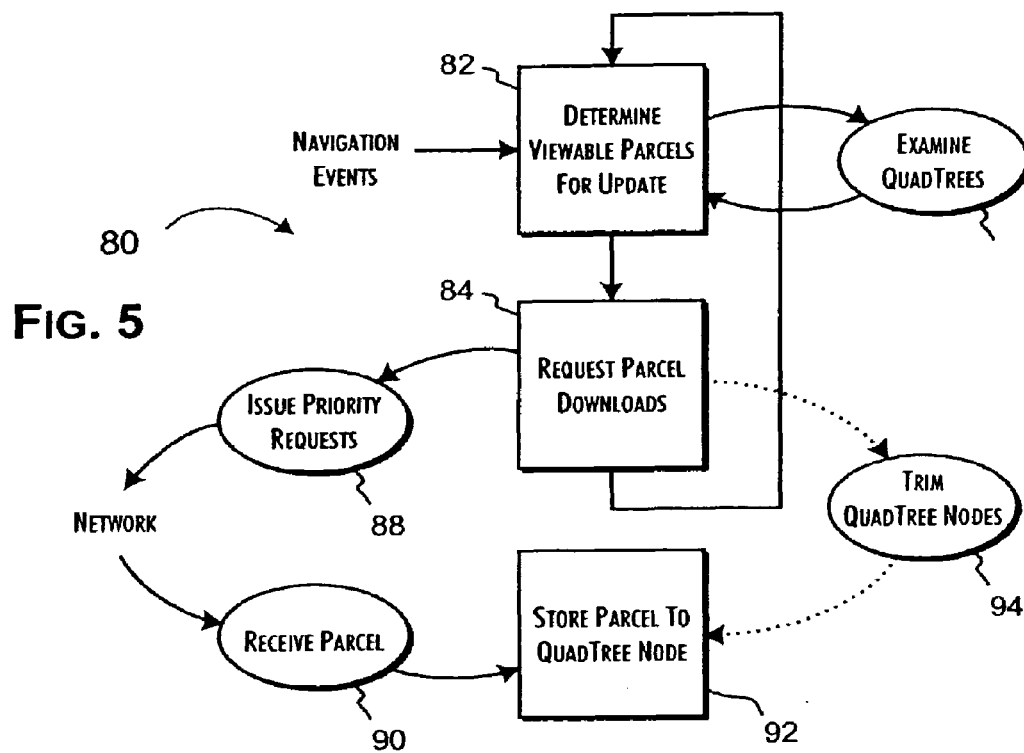
Michalson Declaration Appendices P-EE (all pages).  
 Michalson Declaration with Appendices A-O (all pages).  
 Michael Potmesil, "Maps Alive: Viewing Geospatial Information on the WWW", Computer Networks and ISDN Systems vol. 29, issues 8-13, pp. 1327-1342.  
 Peter Lindstrom, David Koller, William Ribarsky, Larry F. Hodges, Augusto OP den Bosch, Nick Faust, "An Integrated Global GIS and

Visual Simulation System" by P. Lindstrom et al., Tech. Rep. GIT-GVU-97-07, Mar. 1997 (all pages).  
 Curriculum Vitae of William R. Michalson (all pages).  
 Hanan Samet, The Design and Analysis of Spatial Data Structures, University of Maryland (1989) (all pages).  
 B. Fuller and I. Richer, The Magic Project: From Vision to Reality, IEEE Network May/Jun. 1996, pp. 15-25.  
 International Telegraph and Telephone Consultative Committee ("CCITT") Recommendation T.81, Sep. 1992 (all pages).  
 Ken Cabeen & Peter Gent, Image Compression and the Discrete Cosine Transform (all pages).  
 M. Antonini, Image Coding Using Wavelet Transform, IEEE Transactions on Image Processing, vol. 1, No. 2, Apr. 1992 (all pages).  
 Lance Williams, Pyramidal Parametrics, Computer Graphics, vol. 17, No. 3, Jul. 1983, (all pages).  
 OpenGL Standard Version 1.1, Mar. 1997, available: <https://www.opengl.org/documentation/specs/version1.1/glspec1.1/node84.html#SECTION00681100000000000000>, (all pages).  
 H. Hoppe, Progressive Meshes, SIGGRAPH '96: Proceedings of the 23rd annual conference on computer graphics and interactive techniques, pp. 99-108.  
 Boris Rabinovich and Craig Gotsman, Visualization of Large Terrains in Resource-Limited Computing Environments (1997) (all pages).  
 User Datagram Protocol (UDP) (Windows CE 5.0, Microsoft, Available: <https://msdn.microsoft.com/enus/library/ms885773.aspx> [Accessed Apr. 28, 2015] (all pages).  
 OpenGL Standard Version 1.2.1, Apr. 1999, available: <https://www.opengl.org/documentation/specs/version1.2/opengl1.2.1.pdf>, (all pages).  
 George H. Forman and John Zahorjan, "The challenges of mobile computing," Computer vol. 27, No. 4, pp. 38, 47 (Apr. 1994).  
 K. Brown and S. Singh, A Network Architecture for Mobile Computing, INFOCOM '96, Fifteenth Annual Joint Conference of the IEEE Computer Societies, Networking the Next Generation, Proceedings IEEE vol. 3, pp. 1388-139.  
 B. Kreller et al "UMTS: a middleware architecture and mobile API approach," Personal Communications, IEEE, vol. 5, No. 2, pp. 32-38 (Apr. 1998).  
 J. Hansen et al, "Real-time synthetic vision cockpit display for general aviation," AeroSense '99, International Society for Optics and Photonics, 1999, (all pages).  
 Theresa-Marie Rhyne, A Commentary on GeoVRML: A Tool for 3D Representation of GeoReferenced Data on the Web, International Journal of Geographic Information Sciences, issue 4 of vol. 13, 1999, (all pages).  
 Declaration of Prof. William R. Michalson (all pages).  
 Provisional Applications to which the 506 Patent claims priority (all pages).  
 Numbering of Claim Elements of Challenged Claims of the 506 Patent, (all pages).  
 Claim chart illustrating teachings of Potmesil (Ex. 1002) and Hornbacker (Ex. 1003) pertinent to elements of Challenged Claims (all pages).  
 Claim chart illustrating teachings of Rutledge (Ex. 1006), Ligtenberg (Ex. 1005), and Cooper (Ex. 1007) pertinent to elements of Challenged Claims (all pages).  
 Claim chart illustrating teachings of Rutledge (Ex. 1006), Ligtenberg (Ex. 1005), Cooper (Ex. 1007), and Hassan (Ex. 1008) pertinent to elements of Challenged Claims (all pages).  
 Claim chart illustrating teachings of Fuller (App. E) and Hornbacker (Ex. 1003) pertinent to elements of Challenged Claims (all pages).  
 Claim chart illustrating teachings of Yap (App. J) and Rabinovich (App. R) pertinent to elements of Challenged Claims (all pages).

\* cited by examiner







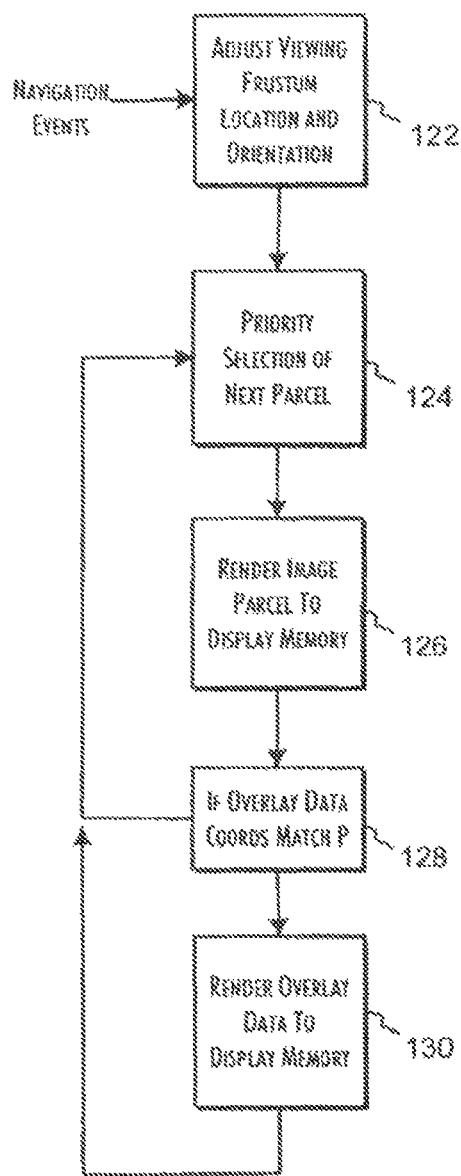
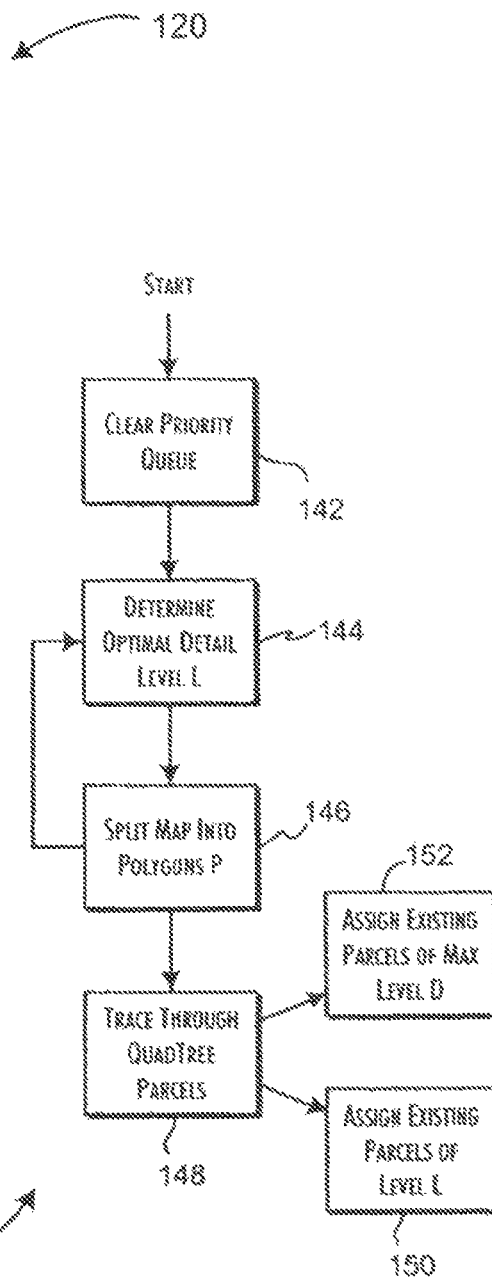
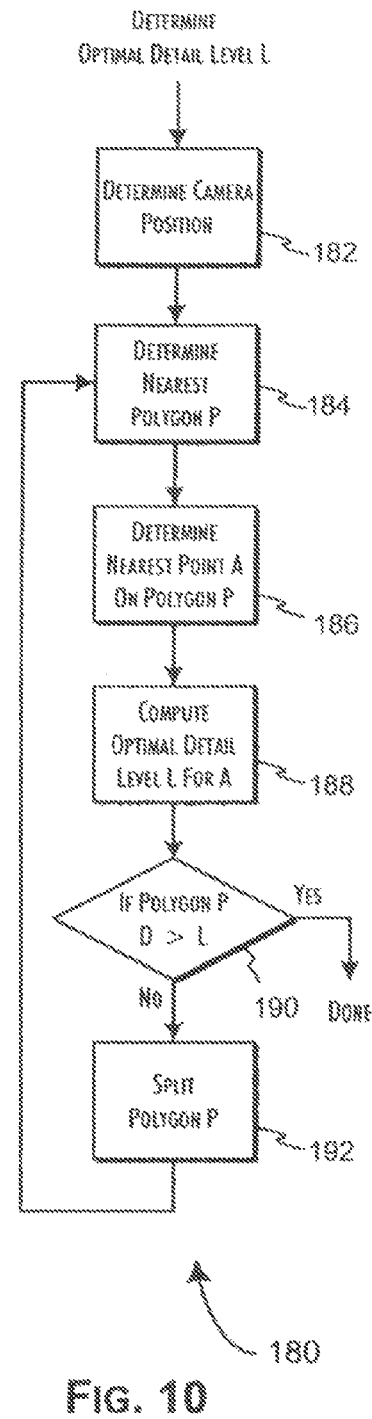
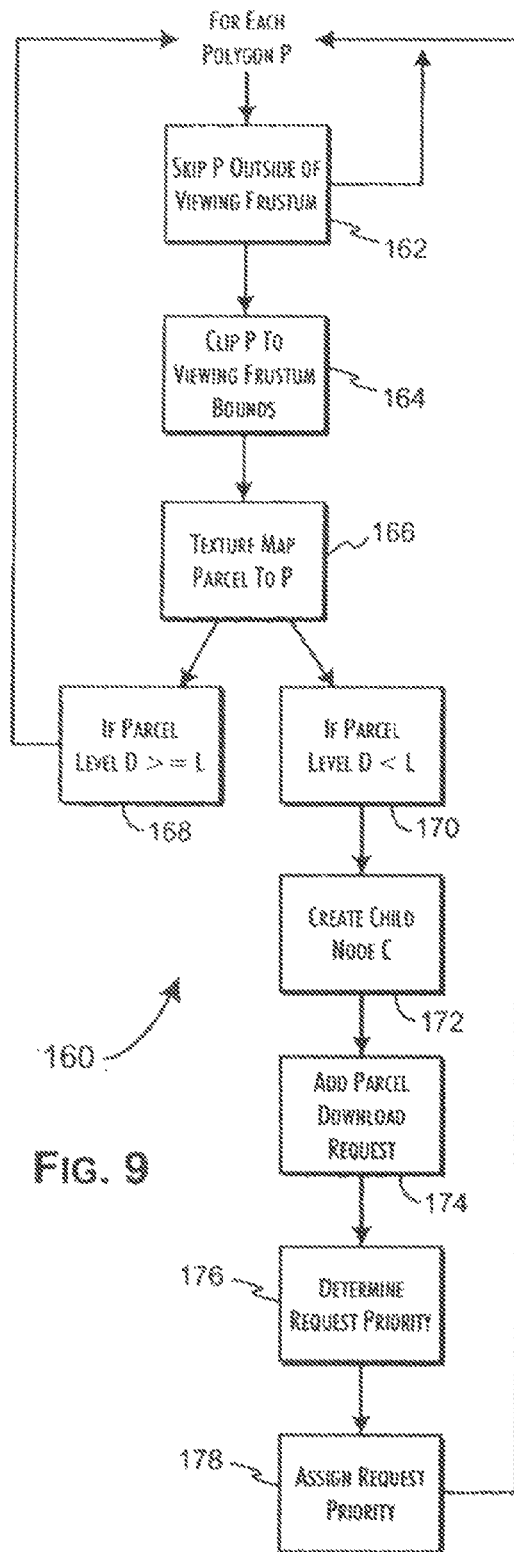


FIG. 7





1

# OPTIMIZED IMAGE DELIVERY OVER LIMITED BANDWIDTH COMMUNICATION CHANNELS

## PRIORITY CLAIMS/RELATED APPLICATIONS

This application is a continuation of and claims priority to U.S. patent application Ser. No. 13/027,929, entitled OPTIMIZED IMAGE DELIVERY OVER LIMITED BANDWIDTH COMMUNICATION CHANNELS, filed on 15 Feb. 2011, now allowed; which is a continuation in part of and claims priority under 35 USC 120 to U.S. patent application Ser. No. 12/619,643 filed on Nov. 16, 2009, now U.S. Pat. No. 7,908,343 which in turn is a continuation of and claims priority under 35 USC 120 to U.S. patent application Ser. No. 10/035,987 filed on Dec. 24, 2001 and entitled "Optimized image delivery over limited bandwidth communication channels" (that now issued on Jan. 5, 2010 as U.S. Pat. No. 7,644, 131) which in turn claims the benefit under 35 USC 119(e) of U.S. Provisional Application Nos. 60/258,488, 60/258,489, 60/258,465, 60/258,468, 60/258,466, and 60/258,467, all filed Dec. 27, 2000, all of which are incorporated herein by reference.

## FIELD

The disclosure is related to network based, image distribution systems and, in particular, to a system and methods for efficiently selecting and distributing image parcels through a narrowband or otherwise limited bandwidth communications channel to support presentation of high-resolution images subject to dynamic viewing frustums.

## BACKGROUND

The Internet and or other network systems may provide a unique opportunity to transmit for example complex images, typically large scale bit-maps, particularly those approaching photo-realistic levels, over large area and or distances. In common application, the images may be geographic, topographic, and or other highly detailed maps. The data storage requirements and often proprietary nature of such images could be such that conventional interests may be to transfer the images on an as-needed basis.

In conventional fixed-site applications, the image data may be transferred over a relatively high-bandwidth network to client computer systems that in turn, may render the image. Client systems may typically implement a local image navigation system to provide zoom and or pan functions based on user interaction. As well recognized problem with such conventional systems could be that full resolution image presentation may be subject to the inherent transfer latency of the network. Different conventional systems have been proposed to reduce the latency affect by transmitting the image in highly compressed formats that support progressive resolution build-up of the image within the current client field of view. Using a transform compressed image transfer function increases the field of the image that can be transferred over a fixed bandwidth network in unit time. Progressive image resolution transmission, typically using a differential resolution method, permits an approximate image to be quickly presented with image details being continuously added over time.

Tzou, in U.S. Pat. No. 4,698,689, describes a two-dimensional data transform system that supports transmission of differential coefficients to represent an image. Subsequent transmitted coefficient sets are progressively accumulated

2

with prior transmitted sets to provide a succeeding refined image. The inverse-transform function performed by the client computer is, however, highly compute intensive. In order to simplify the transform implementation and further reduce the latency of presenting any portion of an approximate image, images are subdivided into a regular array. This enables the inverse-transform function on the client, which is time-critical, to deal with substantially smaller coefficient data sets. The array size in Tzou is fixed, which leads to progressively larger coefficient data sets as the detail level of the image increases. Consequently, there is an inherently increasing latency in resolving finer levels of detail.

An image visualization system proposed by Yap et al., U.S. Pat. No. 6,182,114, overcomes some of the foregoing problems. The Yap et al. system also employs a progressive encoding transform to compress the image transfer stream. The transform also operates on a subdivided image, but the division is indexed to the encoding level of the transform. The encoded transform coefficient data sets are, therefore, of constant size, which supports a modest improvement in the algorithmic performance of the inverse transform operation required on the client.

Yap et al. adds utilization of client image panning or other image pointing input information to support a foveation-based operator to influence the retrieval order of the subdivided image blocks. This two-dimensional navigation information is used to identify a foveal region that is presumed to be the gaze point of a client system user. The foveation operator defines the corresponding image block as the center point of an ordered retrieval of coefficient sets representing a variable resolution image. The gaze point image block represents the area of highest image resolution, with resolution reduction as a function of distance from the gaze point determined by the foveation operator. This technique thus progressively builds image resolution at the gaze point and succeeding outward based on a relatively compute intensive function. Shifts in the gaze point can be responded to with relative speed by preferentially retrieving coefficient sets at and near the new foveal region.

Significant problems remain in permitting the convenient and effective use of complex images by many different types of client systems, even with the improvements provided by the various conventional systems. In particular, the implementation of conventional image visualization systems is generally unworkable for smaller, often dedicated or embedded, clients where use of image visualization would clearly be beneficial. Conventional approaches effectively presume that client systems have an excess of computing performance, memory and storage. Small clients, however, typically have restricted performance processors with possibly no dedicated floating-point support, little general purpose memory, and extremely limited persistent storage capabilities, particularly relative to common image sizes. A mobile computing device such as mobile phone, smart phone, tablet and or personal digital assistant (PDA) is a characteristic small client. Embedded, low-cost kiosk, automobile navigation systems and or Internet enabled I connected TV are other typical examples. Such systems are not readily capable, if at all, of performing complex, compute-intensive Fourier or wavelet transforms, particularly within a highly restricted memory address space.

As a consequence of the presumption that the client is a substantial computing system, conventional image visualization systems also presume that the client is supported by a complete operating system. Indeed, many expect and require an extensive set of graphics abstraction layers to be provided by the client system to support the presentation of the deliv-



ered image data. In general, these abstraction layers are conventionally considered required to handle the mapping of the image data resolution to the display resolution capabilities of the client system. That is, resolution resolved image data provided to the client is unconstrained by any limitation in the client system to actually display the corresponding image. Consequently, substantial processor performance and memory can be conventionally devoted to handling image data that is not or cannot be displayed.

Another problem is that small clients are generally constrained to generally to very limited network bandwidths, particularly when operating under wireless conditions. Such limited bandwidth conditions may exist due to either the direct technological constraints dictated by the use of a low bandwidth data channel or indirect constraints imposed on relatively high-bandwidth channels by high concurrent user loads. Cellular connected PDAs and webphones are examples of small clients that are frequently constrained by limited bandwidth conditions. The conventionally realizable maximum network transmission bandwidth for such small devices may range from below one kilobit per second to several tens of kilobits per second. While Yap et al. states that the described system can work over low bandwidth lines, little more than utilizing wavelet-based data compression is advanced as permitting effective operation at low communications bandwidths. While reducing the amount of data that must be carried from the server to the client is significant, Yap et al. simply relies on the data packet transfer protocols to provide for an efficient transfer of the compressed image data. Reliable transport protocols, however, merely mask packet losses and the resultant, sometimes extended recovery latencies. When such covered errors occur, however, the aggregate bandwidth of the connection is reduced and the client system can stall waiting for further image data to process.

Consequently, there remains a need for an image visualization system that can support small client systems, place few requirements on the supporting client hardware and software resources, and efficiently utilize low to very low bandwidth network connections.

### SUMMARY

Thus, a general purpose of the present invention is to provide an efficient system and methods of optimally presenting image data on client systems with potentially limited processing performance, resources, and communications bandwidth.

This is achieved in the present invention by providing for the retrieval of large-scale images over network communications channels for display on a client device by selecting an update image parcel relative to an operator controlled image viewpoint to display via the client device. A request is prepared for the update image parcel and associated with a request queue for subsequent issuance over a communications channel. The update image parcel is received from the communications channel and displayed as a discrete portion of the predetermined image. The update image parcel optimally has a fixed pixel array size, is received in a single and or plurality of network data packets, and were the fixed pixel array may be constrained to a resolution less than or equal to the resolution of the client device display.

An advantage of the present invention is that both image parcel data requests and the rendering of image data are optimized to address the display based on the display resolution of the client system.

Another advantage of the present invention is that the prioritization of image parcel requests is based on an adaptable parameter that minimizes the computational complexity of

determining request prioritization and, in turn, the progressive improvement in display resolution within the field of view presented on a client display.

A further advantage of the present invention is that the client software system requires relatively minimal client processing power and storage capacity. Compute intensive numerical calculations are minimally required and image parcel data is compactly stored in efficient data structures. The client software system is very small and easily downloaded to conventional computer systems or embedded in conventional dedicated function devices, including portable devices, such as PDAs, tablets and webphones.

Still another advantage of the present invention is that image parcel data requests and presentation can be readily optimized to use low to very low bandwidth network connections. The software system of the present invention provides for re-prioritization of image parcel data requests and presentation in circumstances where the rate of point-of-view navigation exceeds the data request rate.

Yet another advantage of the present invention is that image parcel data rendering is performed without requiring any complex underlying hardware or software display subsystem. The client software system of the present invention includes a bit-map rendering engine that draws directly to the video memory of the display, thus placing minimal requirements on any underlying embedded or disk operating system and display drivers. Complex graphics and animation abstraction layers are not required.

Still another advantage of the present invention is that image parcel block compression is used to obtain fixed size transmission data blocks. Image parcel data is recoverable from transmission data using a relatively simple client decompression algorithm. Using fixed size transmission data blocks enables image data parcels to be delivered to the client in bounded time frames.

A yet further advantage of the present invention is that multiple data forms can be transferred to the client software system for concurrent display. Array overlay data, correlated positionally to the image parcel data and generally insensitive to image parcel resolution, can be initially or progressively provided to the client for parsing and parallel presentation on a client display image view.

### BRIEF DESCRIPTION OF THE DRAWINGS

These and other advantages and features of the present invention will become better understood upon consideration of the following detailed description of the invention when considered in connection with the accompanying drawings, in which like reference numerals designate like parts throughout the figures thereof, and wherein:

FIG. 1 depicts a preferred system environment within which various embodiments of the present invention can be utilized;

FIG. 2 is a block diagram illustrating the preparation of image parcel and overlay data set that are to be stored by and served from a network server system in accordance with a preferred embodiment of the present invention;

FIG. 3 is a block diagram of a client system image presentation system constructed in accordance with a preferred embodiment of the present invention;

FIG. 4 provides a data block diagram illustrating an optimized client image block processing path constructed in accordance with a preferred embodiment of the present invention;

5

FIG. 5 is a process flow diagram showing a main processing thread implemented in a preferred embodiment of the present invention;

FIG. 6 provides a process flow diagram showing a network request thread implemented in a preferred embodiment of the present invention;

FIG. 7 provides a process flow diagram showing a display image rendering thread implemented in a preferred embodiment of the present invention;

FIG. 8 provides a process flow diagram showing the parcel map processing performed preliminary to the rendering of image data parcels in accordance with a preferred embodiment of the present invention;

FIG. 9 provides a process flow diagram detailing the rendering and progressive prioritization of image parcel data download requests in accordance with a preferred embodiment of the present invention; and

FIG. 10 provides a process flow diagram detailing the determination of an optimal detail level for image parcel presentation for a current viewing frustum in accordance with a preferred embodiment of the present invention.

#### DETAILED DESCRIPTION OF ONE OR MORE EMBODIMENTS

The preferred operational environment 10 of the present invention is generally shown in FIG. 1. A network server system 12, operating as a data store and server of image data, is responsive to requests received through a communications network, such as the Internet 14 generally and various tiers of internet service providers (ISPs) including a wireless connectivity provider 16. Client systems, including conventional workstations and personal computers 18 and smaller, typically dedicated function devices often linked through wireless network connections, such as PDAs, webphones 20, and automobile navigation systems, source image requests to the network server 12, provide a client display and enable image navigational input by a user of the client system. Alternately, a dedicated function client system 20 may be connected through a separate or plug-in local network server 22, preferably implementing a small, embedded Web server, to a fixed or removable storage local image repository 24. Characteristically, the client system 18, 20 displays are operated at some fixed resolution generally dependent on the underlying display hardware of the client systems 18, 20.

The image navigation capability supported by the present invention encompasses a viewing frustum placed within a three-dimensional space over the imaged displayed on the client 18, 20. Client user navigational inputs are supported to control the x, y lateral, rotational and z height positioning of the viewing frustum over the image as well as the camera angle of incidence relative to the plane of the image. To effect these controls, the software implemented on the client systems 18, 20 supports a three-dimensional transform of the image data provided from the server 12, 22.

In accordance with the preferred embodiments of the present invention, as generally illustrated in FIG. 2, a network image server system 30 stores a combination of source image data 32 and source overlay data 34. The source image data 32 is typically high-resolution bit-map raster map and or satellite imagery of geographic regions, which can be obtained from commercial suppliers. The overlay image data 34 is typically a discrete data file providing image annotation information at defined coordinates relative to the source image data 32. In the preferred embodiments of the present invention, image annotations include, for example, street, building and landmark names, as well as representative 2 and 3D objects,

6

graphical icons, decals, line segments, and or text and or other characters, graphics and or other media.

The network image server system 30 preferably pre-processes the source image data 32 and or source overlay data 34 to forms preferred for storage and serving by the network server 12, 22. The source image data 32 is preferably pre-processed to obtain a series K.sub.1-N of derivative images of progressively lower image resolution. The source image data 32, corresponding to the series image K.sub.O, is also subdivided into a regular array such that each resulting image parcel of the array has for example a 64 by 64 pixel resolution where the image data has a color or bit per pixel depth of 16 bits, which represents a data parcel size of 8K bytes. The resolution of the series K.sub.1-N of derivative images is preferably related to that of the source image data 32 or predecessor image in the series by a factor of four. The array subdivision is likewise related by a factor of four such that each image parcel is of a fixed 8K byte size.

In the preferred embodiment of the present invention, the image parcels are further compressed and stored by the network server 12, 22. The preferred compression algorithm may implements for example a fixed 4:1 compression ratio such that each compressed and stored image parcel has a fixed 2K byte size. The image parcels are preferably stored in a file of defined configuration such that any image parcel can be located by specification of a K.sub.D, X, Y value, representing the image set resolution index D and corresponding image array coordinate.

In other implementations, the image array dimensions (which as 64×64 above) may be powers of two so that the image array can be used in texture mapping efficiently. To accommodate different data parcel size than the 2 KByte associated with 64×64 pixel parcel dimension described above and other communication protocol and overhead requirements, to accommodate transmission through other than a 3 KByte per second transmission channel, the present invention may use larger compression ratios that takes, for example, a 128×128 or 256×256 pixel parcel dimension and compresses it to meet the 3 KByte per second transmission channel, or other communication bandwidth used to stream the parcel.

The system may also accommodate different and larger data parcel sizes as transmission protocols, compression ratio achieved and micro-architectures of the client computers change. For purposes above, the data content was a pixel array representing image data. Where the data parcel content is vector, text or other data that may subject to different client system design factors, other parcel sizes may be used. Furthermore, the parcel sizes can be different between the server and the client. For example the server may create parcels or hold parcels, for streaming with 256×256 pixel parcel dimension and the client may render them as 64×64. In addition, parcels sizes on different servers may vary from one server to another and from the client side rendering. In the system, each grid is treated as a sparse data array that can be progressively revised to increase the resolution of the grid and thereby the level of detail presented by the grid.

The source overlay data 34 is preferably pre-processed 36 into either an open XML format, such as the Geography Markup Language (GML), which is an XML based encoding standard for geographic information developed by the OpenGIS Consortium (OGC; www.opengis.org), or a proprietary binary representation. The XML/GML representation is preferred as permitting easier interchange between different commercial entities, while the binary representation is preferred as more compact and readily transferable to a client system 18, 20. In both cases, the source overlay data 34 is

pre-processed to contain the annotation data preferably in a resolution independent form associated with a display coordinate specification relative to the source image data 32. The XML, GML or binary overlay data may be compressed prior to storage on the network server 12, 22.

The preferred architecture 40 of a client system 18, 20, for purposes of implementing the present invention, is shown in FIG. 3. The architecture 40 is preferably implemented by software plug-in or application executed by the client system 18, 20 and that utilizes basic software and hardware services provided by the client system 18, 20. A parcel request client 42 preferably implements an HTML client that supports HTML-based interactions with the server 12, 22 using the underlying network protocol stack and hardware network interface provided by the client systems 18, 20. A central parcel processing control block 44 preferably implements the client process and control algorithms. The control block 44 directs the transfer of received image parcels and XML/GML/binary overlay data to a local parcel data store 46. Local parcel data store 46 may also act for example as local cache weather the entire data or part of it is in dynamic and/or static cache. Preferably image data parcels are stored in conventional quad-tree data structures, where tree nodes of depth D correspond to the stored image parcels of a derivative image of resolution KD. The XML/GML/binary overlay data is preferably stored as a data object that can be subsequently read by an XML/GML/binary parser implemented as part of the control block 44.

The control block 44 is also responsible for decompressing and directing the rendering of image parcels to a local display by a rendering engine 48. Preferably, the rendering engine 48 writes to the video memory of the underlying client display hardware relying on only generic graphics acceleration hardware capabilities and may take advantage of more advanced graphics acceleration hardware when available in the client system 18, 20. In general, the relied on capabilities include bit-bit and related bit-oriented functions that are readily supported by current conventional display controller hardware. The rendering engine 48 is optimized to perform image parcel texture mapping without reliance on complex floating point operations, permitting even relatively simple processors to efficiently execute the rendering engine 48. The rendering engine 48 may take advantage of floating point operations when available in the client system 18, 20.

Changes in the viewing frustum are determined from user input navigation commands by a frustum navigation block 50. In the preferred embodiments of the present invention, the input navigation controls are modeled for three-dimensional fly-over navigation of the displayed image. The navigation controls support point-of-view rotation, translation, attitude, and altitude over the displayed image. The effective change in viewing frustum as determined by the frustum navigation block 50 is provided to the control block 44.

The control block 44, based in part on changes in the viewing frustum, determines the ordered priority of image parcels to be requested from the server 12, 22 to support the progressive rendering of the displayed image. The image parcel requests are placed in a request queue 52 for issuance by the parcel request client 42. Preferably, the pending requests are issued in priority order, thereby dynamically reflecting changes in the viewing frustum with minimum latency.

In various implementations of the parcel processing, each data parcel is independently processable by the client system 18, 20, which is enabled by the selection and server-side processing used to prepare a parcel for transmission, thus providing for on-demand real-time parcel processing and

creation on the server side for streaming based on the client request and not only for pre-processed parcel creation for retrieval for streaming from the server. Thus, the system can use both pre-processed parcels on the server and on-demand real-time creation of such parcels on the server side for streaming to the client.

An optimal image parcel data flow 60, as configured for use in the preferred embodiments of the present invention, is shown in FIG. 4. Preferably, the TCP/IP network protocol is used to deliver image parcels to the clients 18, 20. For the preferred embodiments, where network bandwidth is limited or very limited, entire image parcels are preferably delivered in corresponding data packets. This preference maximizes data delivery while avoiding the substantial latency and processing overhead of managing image parcel data split over multiple network packets. Thus, a 2K byte compressed image parcel 62 is delivered as the data payload of a TCP/IP packet 64. Uncompressed, the 8K byte image parcel 62 is recognized as part of the present invention as being within the nominally smallest LI data cache 66 size of conventional microprocessors 68. By ensuring that an uncompressed image parcel fits within the LI cache, the texture map rendering algorithm can execute with minimum memory management overhead, thus optimally utilizing the processing capability of the microprocessor 68. Additionally, the writing of video data as a product of the rendering algorithm is uniform, thereby improving the apparent video stability of the display to the user.

The client architecture 40 preferably executes in multiple process threads, with additional threads being utilized for individual network data request transactions. As shown in FIG. 5, an image parcel management process 80 implements a loop that determines image parcels subject to update 82 and creates corresponding image parcel download requests 84. Navigation events that alter the viewing frustum are considered in part to determine the current field of view. The quad tree data structures are examined 86 to identify viewable image parcels of higher resolution than currently available in the parcel data store 46.

A pool of image request threads is preferably utilized to manage the image parcel download operations. In the preferred embodiments of the present invention, a pool of four network request threads is utilized. The number of pool threads is determined as a balance between the available system resources and the network response latency, given the available bandwidth of the network connection. Empirically, for many wireless devices, four concurrent threads are able to support a relatively continuous delivery of image data parcels to the client 20 for display processing. As image parcels are progressively identified for download, a free request thread is employed to issue 88 a corresponding network request to the server 12, 22. When a network response is received, the corresponding thread recovers 90 the image parcel data. The received image parcel is then stored 92 in a corresponding quad-tree data structure node.

For small clients 20, the available memory for the parcel data store 46 is generally quite restricted. In order to make optimal use of the available memory, only currently viewable image parcels are subject to download. Where the size of the parcel data store 46 is not so restricted, this constraint can be relaxed. In either case, a memory management process 94 runs to monitor use of the parcel data store 46 and selectively remove image parcels to free memory for newly requested image parcels. Preferably, the memory management process 94 operates to preferentially remove image parcels that are the furthest from the current viewing frustum and that have

the highest data structure depth. Preferably child node image parcels are always removed before a parent node parcel is removed.

A preferred network request management process **100** is shown in FIG. 6. The process **100** waits **102** on the existence of a download request in the priority request queue **52**. The process **100** then waits on a network request pool thread to become free **104**. When a network request thread becomes available, the process **100** examines **106** all of the pending requests in the priority request queue **52** and selects **108** the request with the highest assigned priority. Thus, sequentially enqueued requests can be selectively issued out of order based on an independently assigned request priority. The request is then issued **110** and the request management process **100** leaves the request thread waiting on a network response.

FIG. 7 presents a preferred display management process **120**. Event driven user navigation information is evaluated **122** to determine a current viewing frustum location and orientation within a three-dimensional space relative to the displayed image. An algorithmic priority selection **124** of a next image parcel to render is then performed. The selected image parcel is then rendered **126** to the display memory **70**. The rendering operation preferably performs a texture map transform of the parcel data corresponding to the current viewing frustum location and orientation. The overlay data is then parsed or is pre-parsed to determine **128** whether the image coordinates of any overlay annotation correspond to the current image parcel location. If the coordinates match, the overlay annotation is rendered **130** to the video display memory **70**. The process **120** then continues with the next selection **124** of an image parcel to render, subject to any change in the viewing frustum location and orientation.

A preferred implementation of the selection **124** and rendering **126** of image parcels in accordance with the present invention is detailed in FIGS. 8 through 10. Referring first to FIG. 8, any outstanding requests in the priority request queue **52** are preferably cleared **142** in response to a change in the viewing frustum location and orientation. The effective altitude of the viewing frustum and or the resolution of the client display are then used as a basis for determining an optimal level of detail **L** that will be displayed. The detail level **L** value operates as a floor defining the maximum resolution  $K_{sub.L}$  of image data that can be effectively viewed on the client display given the location and or orientation of the viewing frustum. Constraining image parcel requests to the resolution range  $K_{sub.N}$  to  $K_{sub.L}$ , where  $K_{sub.N}$  is the lowest resolution derivative image stored by the network server **12**, **22**, prevents the download and processing of image parcels that cannot provide any perceptible improvement in the displayed image.

As part of the recursive evaluation of the optimal level of detail **L**, the image display space is progressively split **146** by four to one reductions into polygons. The quad-tree data structures holding existing image parcel data in the parcel data store **46** are concurrently traced **148** to establish a correspondence with the polygon map. Where the trace of a quad-tree data structure completes **150** to a node index of **L** for a polygon **P**, the node corresponding image parcel is associated with polygon **P**. The polygon **P** will not be further subdivided and no higher resolution image parcels will be requested for any portion of the image within the area represented by polygon **P**. Where the trace reaches a maximum node index of **D** for a polygon **P'** **152**, where  $N_{ltreq}.D < L$  and **N** is the index of the lowest resolution derivative image stored by the network server **12**, **22**, the image parcel associated with the node is associated with the polygon **P'**. This

polygon **P'** will be subject to further subdivision and progressive requests for image parcels of higher resolution up to the detail level **L**.

Referring now to FIG. 9, a display image is then rendered **160** beginning with the maximum depth polygons previously found. Iterating over the set of maximum depth polygons, any polygons outside of the viewing frustum are skipped **162**. Polygons that are at least partially visible are clipped to the applicable bounds of the viewing frustum **164**. The polygon corresponding image parcel data is then texture mapped **166** into the polygon corresponding coordinates of the video memory **70**. If the node index depth of the rendered image parcel is at least equal to the prior determined optimal detail level **L** **168**, the iteration over the polygons **P** continues.

Where the node index depth is less than the optimal detail level **L** **170**, the polygon **P'** is subdivided into four polygons and correspondingly represented by the creation of four child nodes within the associated quad-tree data structure **172**. Four image parcel download requests are then created **174**.

The download priority associated with each request is determined **176** by execution of a function **S** that operates on a 2D polygon argument **P** and returns a real number representing the request priority. The function argument **P** is a list of real (**x**, **y**) coordinates of the vertices of the current polygon in screen coordinates after being clipped to fit within the current viewing frustum. That is, the function **S** works over general polygons in a two-dimensional space, whose vertices are specified by the series  $\{(x(1), y(1)), (x(2), y(2)), \dots, (x(n), y(n))\}$ . The argument **P** vertices sent to **S** represent the position of the vertices composing each of the polygons, after being clipping to the viewing frustum, viewable within the display space having the fixed resolution [**xRes**, **yRes**]. Thus, the clipped polygons are all within the rectangle  $[O, xRes] \times [O, yRes]$ .

In execution of the function **S**, each of the **P** coordinates is first transformed by linear mapping of the screen coordinate space to the square  $[-1, 1].times. [-1, 1]$  by the operation  $x(i) := (x(i) - xRes/2)/(xRes/2); y(i) := (y(i) - yRes/2)/(yRes/2)$ . The **x** and **y** coordinate values of each vertex  $(x(i), y(i))$  for  $i=1$  to **n** are then transformed by the function  $T(a) = \text{sgn}(a) * \text{pow}(.vertline.a.vertline., d)$ , where the control parameter **d** is a constant in the range (0,1], or equivalently the interval  $0 < d \leq 1$ . The function **S** then returns a real value that is equal to the area covered by the argument polygon **P** vertices subject to the applied coordinate transformation. Thus, the accumulated priority for any image parcel pending download is the sum of the values of returned by the function **S** for each of the viewable polygons that require some part of the image parcel as the source data for texture map rendering of the polygon. The priority operation of the request queue **52** is such that download requests will be issued preferentially for image parcels with the largest priority value.

In accordance with the preferred embodiments of the present invention, the value of the control parameter **d** can be adjusted to ultimately affect the behavior of the function **S** in determining the download request priority. In general, image parcels with lower resolution levels will accumulate greater priority values due to the larger number of polygons that may use a given low resolution image parcel as a rendering data source. Such lower resolution image parcels are therefore more likely to be preferentially downloaded. In accordance with the present invention, this generally assures that a complete image of at least low resolution will be available for rendering.

The control parameter **d**, as applied in execution of the function **S**, well as the area distortion produced by the projection transform also influences the value returned by the

11

function S such that relatively higher-resolution image parcels near the image view point will occasionally achieve a higher priority than relatively remote and partially viewed image parcels of lower resolution. Using values smaller than 1 for the control parameter d results in requests with a higher priority for parcels covering areas near the focal point of the viewer, which is presumed to be the center point of the display space, relative to requests for parcels further from the center point in absolute terms and of the same resolution depth D. Thus, in accordance with the present invention, the priority assigned to image parcel requests effectively influences the order of requests based on the relative contribution of the image parcel data to the total display quality of the image. Empirically, a value of 0.35 for the control parameter d for small screen devices, such as PDAs and webphones has been found to produce desirable results.

The computed priorities of each of the four newly created image parcel requests are then assigned **178** and the requests are enqueued in the priority request queue **52**. The next polygon P is then considered in the loop of the image parcel rendering process **160**.

The preferred algorithm **180** for determining the detail level L value for a given viewing frustum is shown in FIG. **10**. In accordance with the present invention, the optimal detail level L is effectively the limit at which the resolution of image parcel data functionally exceeds the resolution of the client display. Preferably, to determine the optimal detail level L, the viewpoint or camera position of the viewing frustum is determined **182** relative to the displayed image. A nearest polygon P of depth D is then determined **184** from the effective altitude and attitude of the viewpoint. The nearest point A of the polygon P is then determined **186**. The point A may be within the interior or an edge of the polygon P, though most likely be located at a vertex of the polygon P.

The optimum level of detail L at point A is then computed **188** as the base-4 logarithm of the number of pixels on the screen that would be covered by a single pixel from an image parcel of the lowest resolution K.sub.-N image, which is the quad-tree root image and corresponds to an image area covering the entire image map. The point A optimal detail level L is preferably computed analytically from the local value of the Jacobian of the projective transform used to transform the three dimensional image coordinate space to screen coordinates, evaluated at the point A.

Where the depth D of the polygon P is greater than the depth of the computed optimal level of detail L, the detail level L is taken as the optimal detail level L **190**. Thus, through the process **140**, an image parcel or corresponding section of the closest resolution image parcel associated with a parent node in the quad-tree data structure relative to the depth level L will be used as the texture for rendering the polygon P. Conversely, if the depth D is less than that of the optimal detail level L, the polygon P is effectively split into quadrants and the optimal level of detail is reevaluated. The process **180** thus continues iteratively until the optimal detail level L is found.

Thus, a system and methods of optimally presenting image data on client systems with potentially limited processing performance, resources, and communications bandwidth have been described. While the present invention has been described particularly with reference to the communications and display of geographic image data, the present invention is equally applicable to the efficient communications and display of other high resolution information.

In the process implemented by the system described above, data parcels may be selected for sequential transmission based on a prioritization of the importance of the data con-

12

tained. The criteria for the importance of a particular data parcel may be defined as suitable for particular applications and may directly relate to the presentation of image quality, provision of a textual overlay of a low-quality image to quickly provide a navigational orientation, or the addition of topography information at a rate or timing different from the rate of image quality improvement. Thus, image data layers reflecting navigational cues, text overlays, and topography can be composed into data packets for transmission subject to prioritizations set by the server alone and not based on the client system and interactively influenced by the actions and commands provided by the user of the client system. However this also may be influenced based on the nature and type of the client system, and interactively influenced by the actions and commands provided by the user of the client system (FIG. **5**).

In view of the above description of the preferred embodiments of the present invention, many modifications and variations of the disclosed embodiments will be readily appreciated by those of skill in the art. It is therefore to be understood that, within the scope of the appended claims, the invention may be practiced otherwise than as specifically described above.

The invention claimed is:

**1.** A method of retrieving images over a network communication channel for display on a user computing device, the method comprising steps of:

issuing a first request from the user computing device to one or more servers, over one or more network communication channels, the first request being for a first update data parcel corresponding to a first derivative image of a predetermined image, the predetermined image corresponding to source image data, the first update data parcel uniquely forming a first discrete portion of the predetermined image, wherein the first update data parcel is selected based on a first user-controlled image viewpoint on the user computing device relative to the predetermined image;

receiving the first update data parcel at the user computing device from the one or more servers over the one or more network communication channels, the step of receiving the first update data parcel being performed after the step of issuing the first request;

displaying the first discrete portion on the user computing device using the first update data parcel, the step of displaying the first discrete portion being performed after the step of receiving the first update data parcel;

issuing a second request from the user computing device to the one or more servers, over the one or more network communication channels, the second request being for a second update data parcel corresponding to a second derivative image of the predetermined image, the second update data parcel uniquely forming a second discrete portion of the predetermined image, wherein the second update data parcel is selected based on a second user-controlled image viewpoint on the user computing device relative to the predetermined image, the second user-controlled image viewpoint being different from the first user-controlled image viewpoint;

receiving the second update data parcel at the user computing device from the one or more servers over the one or more network communication channels, the step of receiving the second update data parcel being performed after the step of issuing the second request;

displaying the second discrete portion on the user computing device using the second update data parcel, the step

## 13

of displaying the second discrete portion being performed after the step of receiving the second update data parcel;  
 wherein:  
 a series of K1-N derivative images of progressively lower  
 image resolution comprises the first derivative image  
 and the second derivative image, the series of K1-N of  
 derivative images resulting from processing the source  
 image data, series image K0 being subdivided into a  
 regular array wherein each resulting image parcel of the  
 array has a predetermined pixel resolution and a prede-  
 termined color or bit per pixel depth, resolution of the  
 series K1-N of derivative images being related to reso-  
 lution of the source image data or predecessor image in  
 the series by a factor of two, and the array subdivision  
 being related by a factor of two.

2. A method as in claim 1, further comprising:  
 determining the first user-controlled image viewpoint  
 based at least in part on first navigational input of the  
 user computing device; and  
 preparing the first request by a processing control block of  
 the user computing device based at least in part on the  
 first user-controlled image viewpoint.

3. A method as in claim 2, wherein the step of preparing the  
 first request is performed based at least in part on altitude and  
 attitude of the first viewpoint relative to the predetermined  
 image.

4. A method as in claim 2, further comprising:  
 determining the second user-controlled image viewpoint  
 based at least in part on second navigational input of the  
 user computing device; and  
 preparing the second request by a processing control block  
 of the user computing device based at least in part on the  
 second user-controlled image viewpoint.

5. A method as in claim 4, wherein:  
 the step of preparing the first request is performed based at  
 least in part on three-dimensional altitude and attitude of  
 the first viewpoint relative to the predetermined image;  
 and  
 the step of preparing the second request is performed based  
 at least in part on three-dimensional altitude and attitude  
 of the second viewpoint relative to the predetermined  
 image.

6. A method as in claim 5, wherein the predetermined  
 image is an image of a geographic area.

7. A method as in claim 5, wherein:  
 the first navigational input comprises first lateral x dimen-  
 sion position data, first lateral y dimension position data,  
 first z height dimension position data, and first rotational  
 position data; and  
 the second navigational input comprises second lateral x  
 dimension position data, second lateral y dimension  
 position data, second z height dimension position data,  
 and second rotational position data.

8. A method as in claim 5, wherein:  
 the first navigational input comprises first three-dimen-  
 sional coordinate position data and first rotational posi-  
 tion data; and  
 the second navigational input comprises second three-di-  
 mensional position data and second rotational position  
 data.

9. A method as in claim 1, wherein:  
 the first derivative image includes the second derivative  
 image;

## 14

the second derivative image has a higher level of detail than  
 the first derivative image; and the step of issuing the first  
 request is performed before the step of issuing the sec-  
 ond request.

10. A method as in claim 1, wherein:

the second derivative image includes the first derivative  
 image;

the second derivative image has a lower level of detail than  
 the first derivative image; and

the step of issuing the first request is performed before the  
 step of issuing the second request.

11. A method as in claim 1, wherein the first derivative  
 image does not include the second derivative image, and the  
 second derivative image does not include the first derivative  
 image.

12. A method as in claim 1, wherein the first update data  
 parcel comprises first overlay data for the first derivative  
 image.

13. A method as in claim 12, wherein the first overlay data  
 comprises first text annotation relating to at least one of: one  
 or more street names, one or more building names, and one or  
 more landmarks.

14. A method as in claim 12, wherein the first overlay data  
 comprises graphic data representing a three-dimensional  
 object.

15. A method as in claim 12, wherein the first overlay data  
 comprises graphics data describing at least one object in three  
 dimensions.

16. A method as in claim 12, wherein the first overlay data  
 comprises one or more graphical icons.

17. A method as in claim 12, wherein the second update  
 data parcel comprises second overlay data for the second  
 derivative image.

18. A method as in claim 17, wherein the first overlay data  
 and the second overlay data are in a resolution-independent  
 format.

19. A method as in claim 12, wherein the first overlay data  
 comprises first text annotation relating to at least one of: one  
 or more street names, one or more building names, and one or  
 more landmarks.

20. A method as in claim 1, further comprising a step for  
 determining priority of the first request and the second  
 request.

21. A method as in claim 1, wherein:

the steps of issuing the first request and receiving the first  
 update data parcel are part of a first thread;

the steps of issuing the second request and receiving the  
 second update data parcel are part of a second thread;  
 and

the first thread and the second thread are executed at least  
 in part concurrently.

22. A method as in claim 1, further comprising:

issuing a third request from the user computing device to  
 the one or more servers, over the one or more network  
 communication channels, the third request being for a  
 third update data parcel corresponding to a third deriva-  
 tive image of the predetermined image, the third update  
 data parcel uniquely forming a third discrete portion of  
 the predetermined image;

receiving the third update data parcel at the user computing  
 device from the one or more servers over the one or more  
 network communication channels;

issuing a fourth request from the user computing device to  
 the one or more servers, over the one or more network  
 communication channels, the fourth request being for a  
 fourth update data parcel corresponding to a fourth  
 derivative image of the predetermined image, the fourth

15

update data parcel uniquely forming a fourth discrete portion of the predetermined image; and  
receiving the fourth update data parcel at the user computing device from the one or more servers over the one or more network communication channels; 5

wherein:

the steps of issuing the first request and receiving the first update data parcel are part of a first thread;

the steps of issuing the second request and receiving the second update data parcel are part of a second thread; 10

the steps of issuing the third request and receiving the third update data parcel are part of a third thread;

the steps of issuing the fourth request and receiving the fourth update data parcel are part of a fourth thread; and 15

the first thread, the second thread, the third thread, and the fourth thread are executed at least in part concurrently.

**23.** A method as in claim 1, wherein the user computing device is a mobile device.

**24.** A method as in claim 1, wherein the one or more servers comprise at least two servers. 20

**25.** A method as in claim 1, wherein each image parcel is of a fixed byte size.

\* \* \* \* \*

16